Implementing Parallel and Independent Movements for a Social Robot's Affective Expressions

Hannes Ritschel, Thomas Kiderle and Elisabeth André
Human-Centered Artificial Intelligence, Augsburg University
Augsburg, Germany
{hannes.ritschel, thomas.kiderle, elisabeth.andre}@uni-a.de

***Abstract*—The design and playback of natural and believable movements is a challenge for social robots. They have several limitations due to their physical embodiment, and sometimes also with regard to their software. Taking the example of the expression of happiness, we present an approach for implementing parallel and independent movements for a social robot, which does not have a full-fledged animation API. The technique is able to create more complex movement sequences than a typical sequential playback of poses and utterances and thus is better suited for expression of affect and nonverbal behaviors.**

***Index Terms*—social robots, movement, animation, emotion**

I. Motivation

Natural and believable movements are a key challenge for social robots. Especially (but not exclusively) when it comes to the expression of emotional and affective nonverbal behaviors inspired by human-human communication, we have very high expectations towards the behaviors of anthropomorphized agents. We learn how to communicate and interpret human social signals, such as gestures and facial expressions, for decades in our lives. Thus, the perceived naturalness of social robot animations[1] constitutes a key challenge, both in terms of the hardware and software, but also in terms of artistic skills to create pleasing movements.

Animation studios have mastered this problem for decades. In 1981, Johnston and Thomas published the principles for traditional animation [1] based on what has been perfected in Disney animations for years. With computer graphics being a de facto standard in today's movie production, interactive media and everyday life, these principles have also been applied to this domain [2] by many big animation studios and ever since serve as elementary tools for every aspiring animator, who aims to produce an "illusion of life" [1].

Traditionally, these principles are applied by hand, which requires much practice and artistic skills. Motion capturing has become an important technique for transferring human movements as adquate as possible to virtual counterparts. In addition, neural animation generation approaches become more and more popular, too. For example, they allow for animating a virtual violinist's hand [3], to synthesize emotional expressions of a virtual 3D face [4] or to produce appropriate gestures for a 3D skeleton based on a given text [5]. Neural movement techniques were also applied to social robots, e.g. to generate co-speech gestures for a NAO robot [6] and for generation of affective robot movements [7]. Karg et al. [8] provide an overview of affect-expressive movement generation, including virtual agents and robots. Apart from linguistic contents, compelling multimodal robot behaviors with animations are important when expressing personality [9]–[14], jokes and humor [15]–[19] and emotions [20], [21].

While traditional hand-drawn animation, computer graphics and virtual agents provide the necessary tools and flexibility to animate, manipulate and deform characters to almost any shape, social robots are much more limited due to their physical embodiment. Nevertheless, a robot's movement abilities have a huge impact on the degree to which it is able to portray emotions and personality.

Regardless of whether movements are scripted by hand or produced with a generative approach, a corresponding software for playing back these movements on the robot's hardware is essential. However, some robotic platforms do not provide full-fledged movement abilities out of the box. We illustrate major benefits of keyframe animation based on the basic emotion happiness, how to implement it with the Universal Robot Body Interface (URBI) for the Reeti robot and give basic tips on how to improve animation playback from a technical point of view.

II. Implementing Keyframe Animation with URBI

We illustrate the implementation of keyframe animation with URBI based on the Reeti robot, which has an extraterrestrial face. Since URBI is a standard, which is included in several robotic products, the following technique should be applicable to other robotic platforms, too.

In contrast to the Behavior Markup Language (BML) [22], which defines behaviors on a higher level of abstraction (such as planning which gestures to perform in which order, gaze targets, head and body movements), URBI scripts are more platform-specific, which realize these movements in terms of precise motor instructions. For example, a BML solver for the Reeti robot could generate corresponding URBI scripts for realizing each chunk of the BML sequence.

The Reeti robot has motors in its head for generating facial expressions. Its software has several limitations with regards to animation creation and playback, which, however, can be

[1]In this paper, we use the terms *movement* and *animation* interchangeable: in the context of robots, animation also means the movement or manipulation of a robot's actuators over time. This also includes non-motorized actuators, such as LED lights or Text-To-Speech (TTS) output.

bypassed with URBI scripts (see below). By default, the manufacturer's Application Programming Interface (API) provides functionality for setting the robot's pose, which causes all motors to move at the same time into a new position. In addition, one can script a reduced form of keyframe animation with software on the robot itself. These animations cannot be modified during runtime of an application. However, in typical social robot applications, animations must be created on-the-fly depending on the user's input, e.g. to update the text of an utterance or to control the degree how strong an emotion should be expressed. Moreover, a robot should be able to move actuators independently of the others, such as in natural human or animal movements.

A. "Pose by Pose" Animation

The left part of Figure 1 illustrates the limitations of the robot's API. Robot animations have to be realized by playing different poses one after the other, which we call "pose by pose" animation in the following. The example shows a sequence of pose and Text-To-Speech (TTS) commands, which, in combination, aims to express happiness. Smiling is used as facial expression to support the utterance "I am so happy!" In addition, the robot's eyes blink two times. The robot is blocked during playback of poses and talking, which means that no other movement or TTS output can happen in parallel. The resulting movement is choppy and takes much time because one movement happens after the other and talking cannot be combined with movements at the same time. Thus, there are mandatory breaks between utterances as soon as movements happen inbetween. In addition, each command introduces a network delay, which is primarily noticeable when working in wireless networks (see section III). Speech output is produced by the robot's internal TTS system or the commercial Cerevoice TTS, which also supports emotional output for some voices.

B. Keyframe Animation

Keyframe animation is a technique from traditional animation. The *keyframes* are the most important frames, which give the animator an idea of the extreme poses of a character throughout the animation. They are drawn first. The *in-between* frames, which interpolate the keyframes, are drawn afterwards. In computer animation and robotics, this is done by interpolation algorithms.

An animation *timeline* consists of several *tracks*. There exists one track for each of the robot's actuators (motors, LEDs, text output). *Keyframes* are placed on each track to define the position, value or text of the actuator at a specific point in time. *Interpolation* is used to calculate the values in between the keyframes. The tracks are independent among each other, i.e., keyframes on different tracks do not need to share the same time offsets, since interpolation algorithms can calculate the values for each point in time automatically.

The Reeti robot's API does not provide this type of animation. However, it can be implemented by generating corresponding URBI scripts to create complex animations in real-time as follows (see Listing 1, too):

- Each track is encapsulated in curly brackets. Tracks are separated with an ampersand (`&`).
- Keyframes are set with `motor=value smooth: offset`, where *motor* is the actuator's identifier and *value* is its value at time *offset* (offset is relative to the preceding keyframe).
- Keyframes are separated with a vertical stroke (`|`).
- LED keyframes can be set with the `changeLedColorRGB` function in combination with a preceding `sleep(offset)` command, with *offset* being the keyframe time.
- The `say()` function is used for TTS output.
- The resulting script is terminated with a vertical stroke (`|`).

C. Example

Listing 1 shows a small excerpt of an URBI script for the Reeti robot. For example, the *neckTilt* motor is set to value 50 after 0.5 seconds. 1.5 seconds later, its value is set to 75. *rightEyeTilt* is animated in parallel. Task of the URBI console is to calculate the in-between values and to translate the instructions into hardware movements. The function *changeLedColorRGB* is used to set the robot's LED in combination with the *sleep* function, which sets the offset on the timeline. The robot starts its utterance "I am so happy!" at the beginning of the animation. After sending the resulting script to the robot's URBI console, the robot starts playing back the instructions. In contrast to the "pose-by-pose" approach, the animation consists of only one single command, which reduces delays caused by network communication and results in more fluent movements. The robot blocks until the script is finished.

Listing 1: A small part of an URBI script implementing keyframe animation with independent, parallel movement of motors, text output and LED color.

```
{ Global.servo.neckTilt=50 smooth: 0.5 | Global.
servo.neckTilt=75 smooth: 1.5 } & { Global.servo.
rightEyeTilt=40 smooth: 0.75 | Global.servo.
rightEyeTilt=19 smooth: 1.0 } & { sleep(1.0) |
Global.servo.changeLedColorRGB(2,1023,1023,0,1) }
& { Global.tts.say("\\language=English \\voice=
Simon \\volume=50 I am so happy!") } |
```

Figure 2 shows stills[2] of the "pose by pose" and keyframe animation approach from Figure 1. In the top row, all movements are limited to distinct poses, which are processed one after the other (e.g. 1-3 blink, 4 smile, talk, 5 return to neutral position, 6-8 blink). In the bottom row, keyframe animation allows to blend movements into each other (e.g. 1 start ear movement and speaking, 2 set LED and start eye movement, 3 start blink and start head movement, 4-5 continue head movement, blink already finished, 6-7 blink, 8 stop eye movement, 9 stop ear movement).

[2]Full video: https://archive.org/details/robot-keyframe-movement

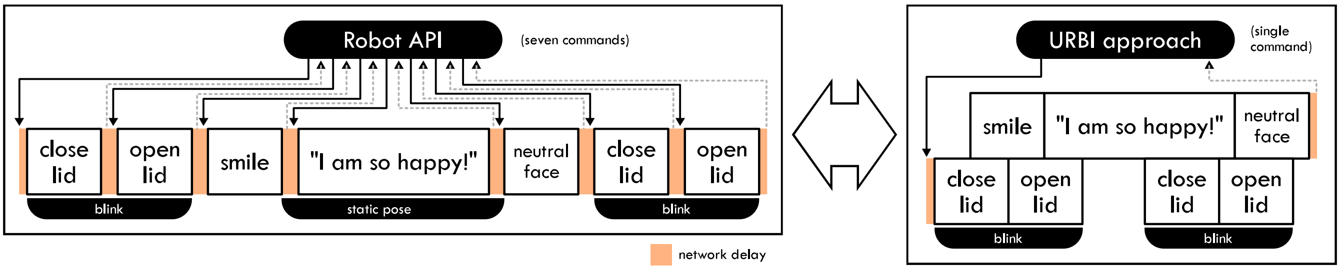

Fig. 1: Two approaches for generating movements during runtime of a Reeti robot application. Left: the Reeti robot's API allows for setting poses and talking one after the other, but not in parallel. Right: URBI can be used for parallel and independent movements (keyframe animation). The orange bars mark delays usually introduced by wireless network communication.

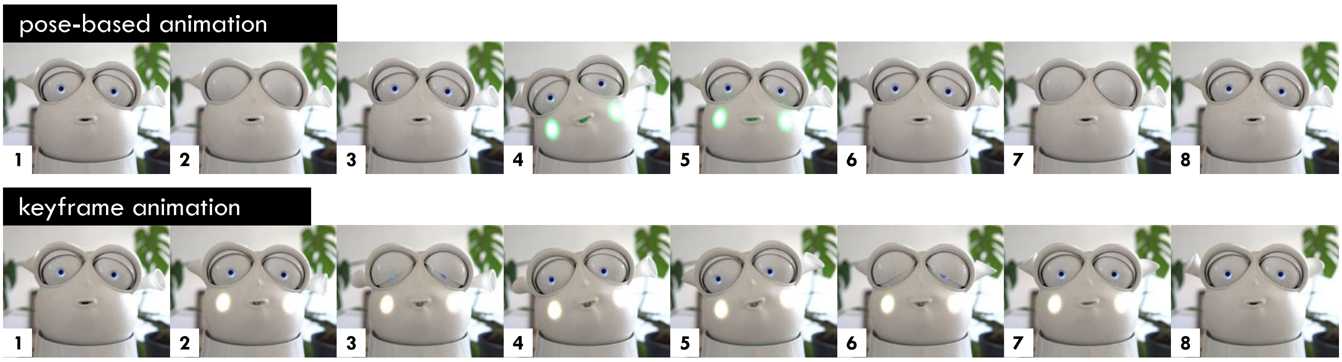

Fig. 2: Stills of the animation from Figure 1. Top: the robot presents several poses and its utterance sequentially without parallel movements. Bottom: several actuators move in parallel when using keyframe animation via URBI scripts (e.g. movement of the ears, blink during head movement, talk while moving, etc.).

As illustrated on the right side of Figure 1, the approach allows to create e.g. robot utterances, which are augmented with parallel movements, such as eye blinks, facial expressions or gaze behavior at the same time. In addition, the resulting movements take less time and appear more natural, because the robot does not remain still for a longer period of time. This is also helpful to avoid lengthy interactions, where users have to wait for the robot's next utterance because it is moving. While writing URBI code by hand is possible, it is reasonable to write a converter to create these scripts based on a custom API or data structures (e.g. for generative animation approaches).

III. General Tips & Tricks

Using URBI scripts allows for parallel and independent movement of all actuators of the Reeti robot. It makes animation more fluid as compared to several command invocations via the robot manufacturer's API. This is due to the fact that all movements of the animation are combined into one single command, which prevents additional network delays. In addition, we observed the following tips and tricks:

1) In order to get as fluid and realistic movement as possible, one should reduce delays as much as possible. This includes e.g. network overhead. When controlling the robot remotely, wireless networks should be avoided and cables should be used for minimal delays. If possible, one can also run the software on the robot's hardware itself.
2) Typically, human movements happen fast. However, one basic mistake when creating movements by hand is that timing is not correct, e.g. the speed is much too slow, which does not look very natural. Looking at references, e.g. video recordings, can help. For example, a robot saying "I am so happy!" with very slow movement might not be very convincing. In contrast, a robot saying "I am so sad." might benefit from slower movements.
3) Humans and animals are always in motion and never remain still. There should always be a subtle motion, otherwise, the robot might look static and lifeless in the long run. This can be addressed by implementing basic idle movements, gaze behaviors, eye blinks or saccades [23]. These animations can be generated automatically with custom program logic.

IV. Future Work

In future work, one option to evaluate this technical contribution is to design several animations (e.g. exemplary emotions, such as done in [20]) with both techniques: one set of animations based on the "pose by pose" approach and the same set of animations based on the proposed keyframe animation/URBI script technique. Then, study participants observe both variants in randomized order for each emotion without being told which emotion it is intended to express. They assign it an emotion, rate which variant they prefer and rate for each variant the perceived naturalness of the movement. In combination, this will also give insights about human individual and subjective differences with regard to perceived emotional expression and naturalness of the animation. Another interesting aspect to investigate is a potential channel mismatch when the TTS output might express one (or no specific) emotion, and the animation another one. Thus, it might be beneficial to evaluate the movements with and without text and to address the TTS quality in specific.

V. Conclusion

In this paper, we presented an approach for implementing parallel and independent movements for a social robot without a full-fledged animation API. Based on the emotion happiness, we demonstrated how this technique can be used to augment a positive utterance with additional facial expression and movements. In contrast to the sequential presentation of poses and text, the proposed approach is able to create complex animations during runtime of an interaction, which allows for generative and automated movements. Moreover, it allows to reduce the time required for playing back combined sequences of utterances and movements because of parallel execution and reduced network delays. On top of that, we presented some tips and tricks to further improve animation playback and design. Thus, the proposed technique is an important step for designing and implementing expressive robot movements, such as affective expressions, for the Reeti robot, which might also be applicable to other robots with an URBI interface.

Acknowledgments

This research was funded by the European Union PRESENT project, grant agreement No 856879.

References

[1] O. Johnston and F. Thomas, *The illusion of life: Disney animation*. Disney Editions New York, 1981.

[2] J. Lasseter, "Principles of traditional animation applied to 3d computer animation," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*. ACM, 1987, pp. 35–44.

[3] J. Kim, F. Cordier, and N. Magnenat-Thalmann, "Neural network-based violinist's hand animation," in *Computer Graphics International Conference, CGI 2000, Geneva, Switzerland, June 19-24, 2000*. IEEE Computer Society, 2000, pp. 37–41.

[4] P. Hong, Z. Wen, and T. S. Huang, "Real-time speech-driven face animation with expressions using neural networks," *IEEE Trans. Neural Networks*, vol. 13, no. 4, pp. 916–927, 2002.

[5] T. Kucherenko, P. Jonell, S. van Waveren, G. E. Henter, S. Alexandersson, I. Leite, and H. Kjellström, "Gesticulator: A framework for semantically-aware speech-driven gesture generation," in *ICMI '20: International Conference on Multimodal Interaction, Virtual Event, The Netherlands, October 25-29, 2020*. ACM, 2020, pp. 242–250.

[6] Y. Yoon, W. Ko, M. Jang, J. Lee, J. Kim, and G. Lee, "Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots," in *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*. IEEE, 2019, pp. 4303–4309.

[7] M. Suguitan, M. Bretan, and G. Hoffman, "Affective robot movement generation using cyclegans," in *14th ACM/IEEE International Conference on Human-Robot Interaction, HRI 2019, Daegu, South Korea, March 11-14, 2019*. IEEE, 2019, pp. 534–535.

[8] M. Karg, A. Samadani, R. Gorbet, K. Kühnlenz, J. Hoey, and D. Kulic, "Body movements for affective expression: A survey of automatic recognition and generation," *IEEE Trans. Affect. Comput.*, vol. 4, no. 4, pp. 341–359, 2013.

[9] H. Kim, S. S. Kwak, and M. Kim, "Personality design of sociable robots by control of gesture design factors," in *The 17th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2008, Munich, Germany, August 1-3, 2008*. IEEE, 2008, pp. 494–499.

[10] A. Aly and A. Tapus, "Towards an intelligent system for generating an adapted verbal and nonverbal combined behavior in human–robot interaction," *Autonomous Robots*, vol. 40, no. 2, pp. 193–209, 2015.

[11] C. Hu, M. A. Walker, M. Neff, and J. E. F. Tree, "Storytelling agents with personality and adaptivity," in *Intelligent Virtual Agents - 15th International Conference, IVA 2015, Delft, The Netherlands, August 26-28, 2015, Proceedings*, ser. Lecture Notes in Computer Science, vol. 9238. Springer, 2015, pp. 181–193.

[12] H. Ritschel, T. Baur, and E. André, "Adapting a robot's linguistic style based on socially-aware reinforcement learning," in *26th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2017, Lisbon, Portugal, August 28 - Sept. 1, 2017*, 2017, pp. 378–384.

[13] H. Ritschel, A. Seiderer, K. Janowski, S. Wagner, and E. André, "Adaptive linguistic style for an assistive robotic health companion based on explicit human feedback," in *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2019, Island of Rhodes, Greece, June 5-7, 2019*, 2019, pp. 247–255.

[14] T. Kiderle, H. Ritschel, K. Janowski, S. Mertes, F. Lingenfelser, and E. André, "Socially-aware personality adaptation," in *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, 2021.

[15] H. Knight, "Eight lessons learned about non-verbal interactions through robot theater," in *Social Robotics - Third International Conference, ICSR 2011, Amsterdam, The Netherlands, November 24-25, 2011. Proceedings*, 2011, pp. 42–51.

[16] K. Katevas, P. G. Healey, and M. T. Harris, "Robot comedy lab: experimenting with the social dynamics of live performance," *Frontiers in psychology*, vol. 6, 2015.

[17] H. Ritschel and E. André, "Shaping a social robot's humor with natural language generation and socially-aware reinforcement learning," in *Proceedings of the Workshop on NLG for Human–Robot Interaction*, 2018, pp. 12–16.

[18] H. Ritschel, I. Aslan, D. Sedlbauer, and E. André, "Irony man: Augmenting a social robot with the ability to use irony in multimodal communication with humans," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '19. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 86–94.

[19] H. Ritschel, T. Kiderle, K. Weber, F. Lingenfelser, T. Baur, and E. André, "Multimodal joke generation and paralinguistic personalization for a socially-aware robot," in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection*. Springer International Publishing, 2020, pp. 278–290.

[20] M. Häring, N. Bee, and E. André, "Creation and evaluation of emotion expression with body movement, sound and eye color for humanoid robots," in *20th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2011, Atlanta, Georgia, USA, July 31 - August 3, 2011*, 2011, pp. 204–209.

[21] H. Ritschel, I. Aslan, S. Mertes, A. Seiderer, and E. André, "Personalized synthesis of intentional and emotional non-verbal sounds for social robots," in *8th International Conference on Affective Computing and Intelligent Interaction, ACII 2019, Cambridge, United Kingdom, September 3-6, 2019*. IEEE, 2019, pp. 1–7.

[22] H. H. Vilhjálmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. N. Marshall, C. Pelachaud, Z. Ruttkay, K. R. Thórisson, H. van Welbergen, and R. J. van der Werf, "The behavior markup language: Recent developments and challenges," in *Intelligent Virtual Agents, 7th International Conference, IVA 2007, Paris, France, September 17-19, 2007, Proceedings*, ser. Lecture Notes in Computer Science, vol. 4722. Springer, 2007, pp. 99–111.

[23] K. Ruhland, S. Andrist, J. B. Badler, C. E. Peters, N. I. Badler, M. Gleicher, B. Mutlu, and R. McDonnell, "Look me in the eyes: A survey of eye and gaze animation for virtual agents and artificial systems," in *35th Annual Conference of the European Association for Computer Graphics, Eurographics 2014 - State of the Art Reports, Strasbourg, France, April 7-11, 2014*. Eurographics Association, 2014, pp. 69–91.